CRYPTORA

RSA BASED ENCYPTION - DECRYPTION TOOL

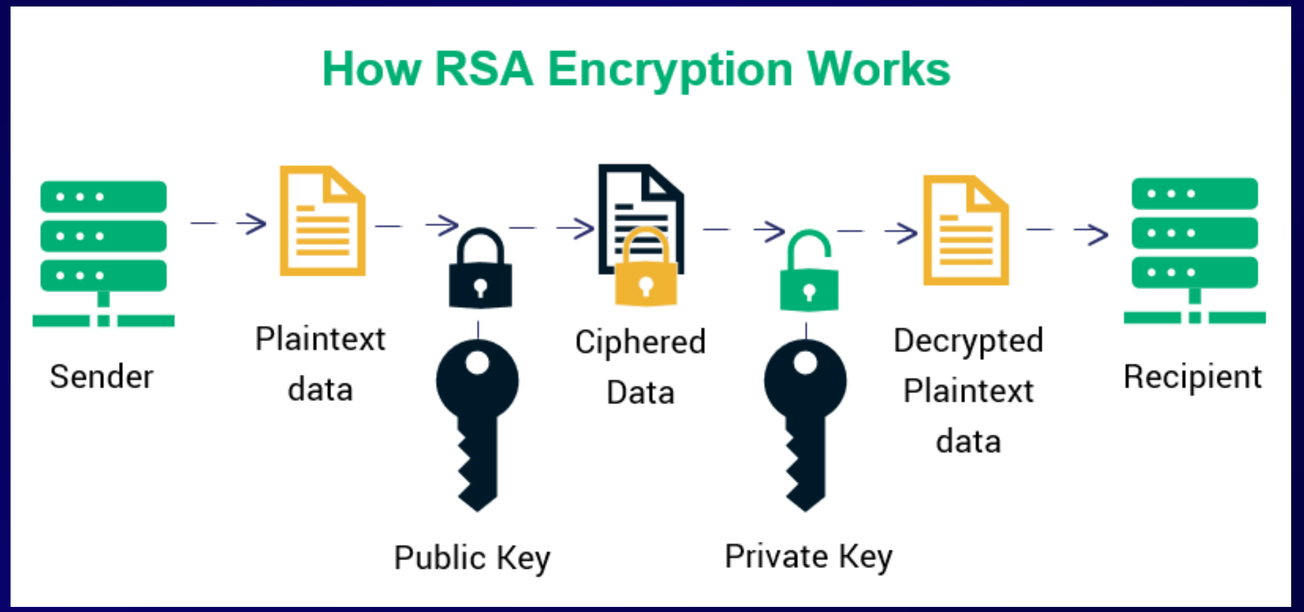
ASSYMETRIC KEY CRYPTOGRAPHY

Submitted By - Chhavi Rohilla
Submitted To - Abhilasha Mam

ASYMMETRIC KEY CRYPTOGRAPHY,

IS A METHOD OF ENCRYPTING AND DECRYPTING DATA USING A PAIR OF MATHEMATICALLY RELATED KEYS.

PUBLIC KEY: SHARED OPENLY WITH ANYONE
PRIVATE KEY: KEPT CONFIDENTIAL



p,q -> large prime numbers

n -> = p*q

phiN -> (p-1)(q-1) #totient

e -> encryption exponent # coprime with phiN && 1 < e <= phiN

d -> decryption exponent #e^d (mod phiN) = 1

c -> cipher text

m -> main message

Public key (n, e)

Private key (n, d)

Encyption:

UserA ----------(n,e)----------> UserB

<----------c = m^e mod (n)----------

Decyption:

UserA <---------- c ---------- UserB

(n, e, d)

m=c^d mod (n)

TECH STACK :

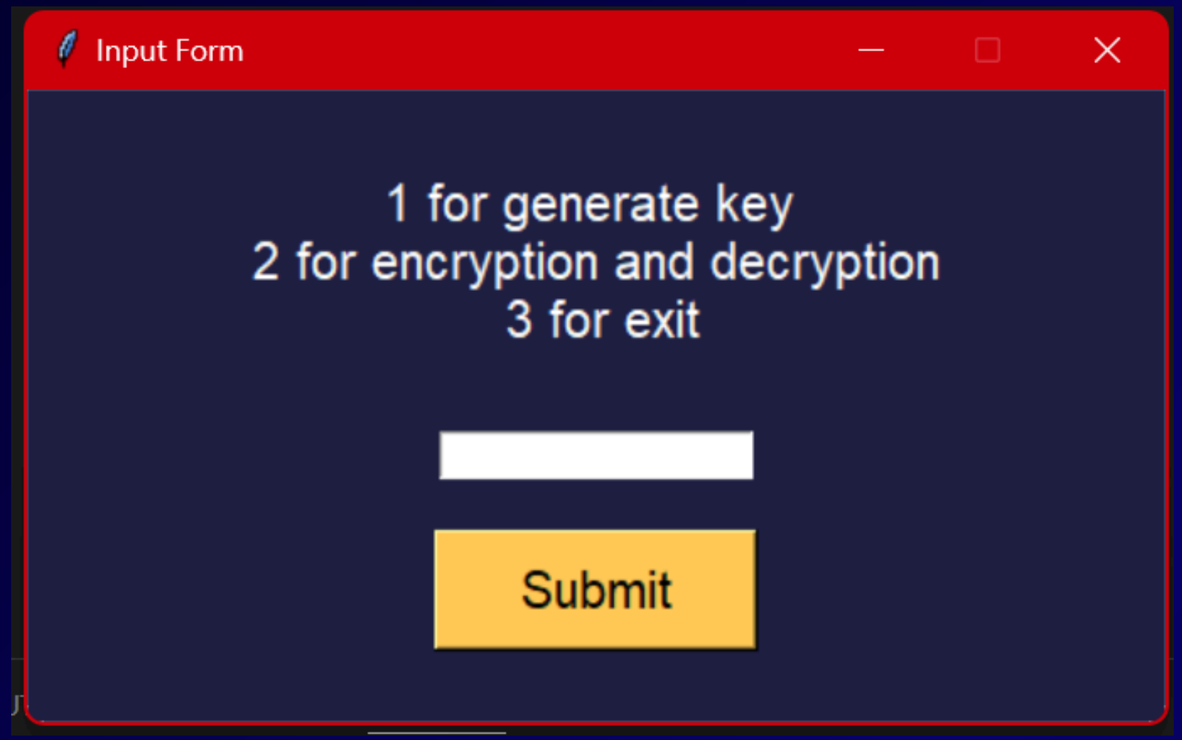
-> PYTHON SCRIPT
-> TKINTER MODULE FOR GUI INTERFACE

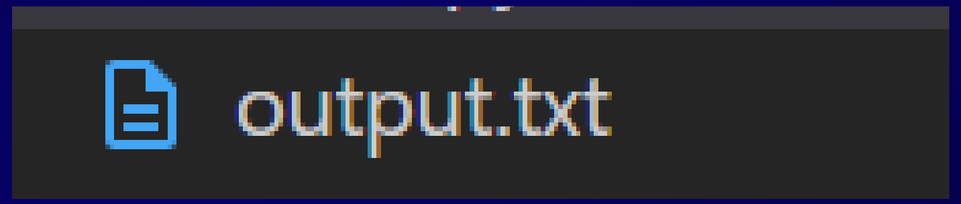
MATH STACK :

-> EUCLIDEAN ALGORITHM FOR GCD
-> EXTENDED EUCLIDEAN ALGORITHM MOD INV

WHERE IT HELPS:

CAREFREE EXCHANGE OF PASSWORDS AND INFORMATION.

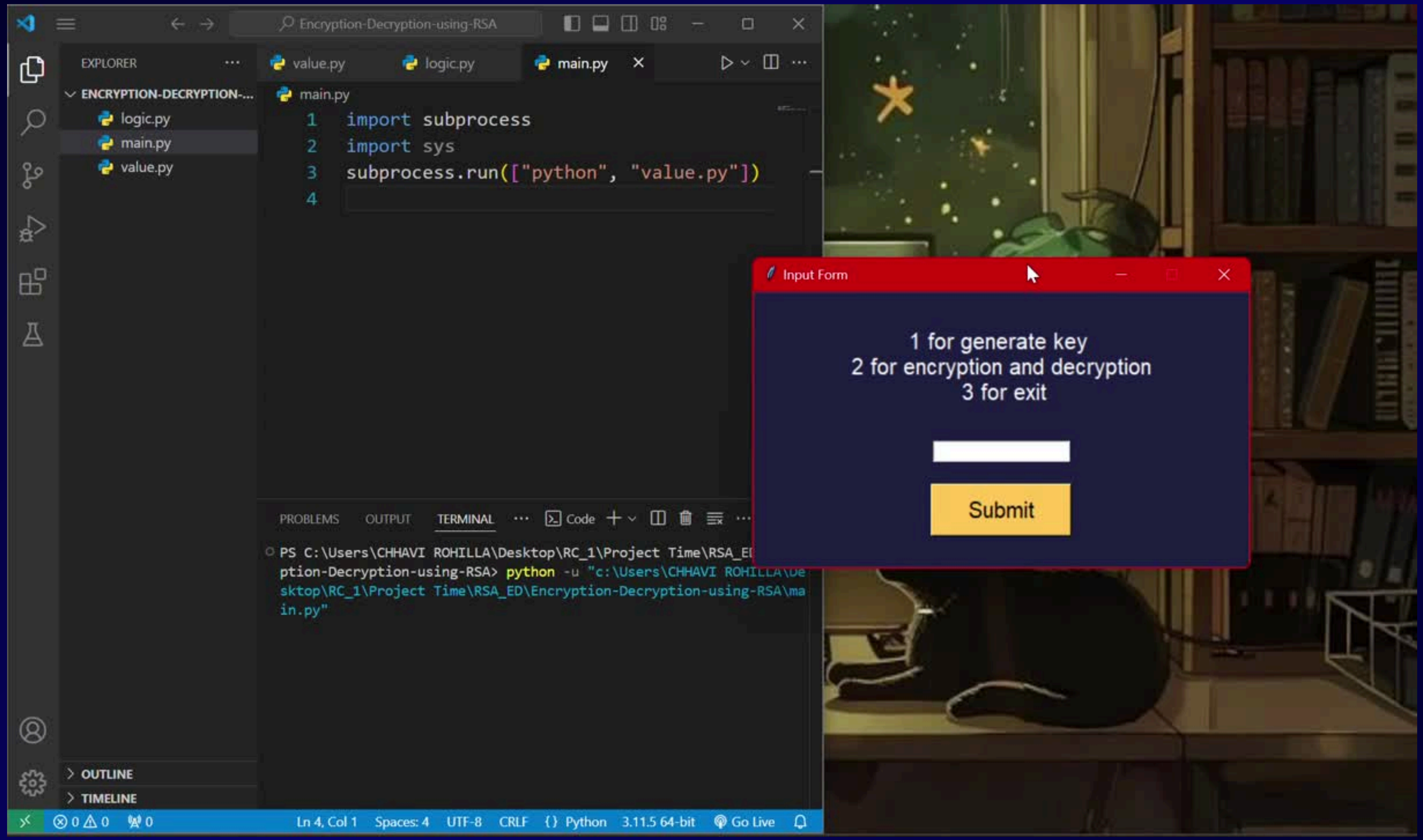

Input Form
1 for generate key
2 for encryption and decryption
3 for exit
Submit


output.txt


Input Form
enter the choice
1 for encryption 2 for decryption
give


RSA Encryption
RSA Encryption/Decryption Tool
Encryption Key:
N
Message:
Encrypt
Encrypted Message:


RSA Decryption
RSA Encryption/Decryption Tool
Encryption Key:
N
Decryption Key:
Message:
Decrypt
Decrypted Message:


Encryption-Decryption-using-RSA
EXPLORER
value.py
logic.py
main.py
ENCRYPTION-DECRYPTION-...
logic.py
main.py
value.py
main.py
1 import subprocess
2 import sys
3 subprocess.run(["python", "value.py"])
4
PROBLEMS
OUTPUT
TERMINAL
Code
PS C:\Users\CHHAVI ROHILLA\Desktop\RC_1\Project Time\RSA_ED
ption-Decryption-using-RSA> python -u "c:\Users\CHHAVI ROHILLA\De
sktop\RC_1\Project Time\RSA_ED\Encryption-Decryption-using-RSA\ma
in.py"
OUTLINE
TIMELINE
Ln 4, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.11.5 64-bit
Go Live
Input Form
1 for generate key
2 for encryption and decryption
3 for exit
Submit

THANK YOU